NASA-TM-112338

✓

Presented at the Fourth Annual ISA Workshop on Robotics and Expert Systems,
Palo Alto, CA, August 2-4, 1989
Reproduced on pages 267-272 of the Proceedings

Higher-Order Neural Networks for Invariant Pattern Recognition

Max B. Reid, Lilly Spirkovska, and Ellen Ochoa
Intelligent Systems Technology Branch
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

A higher-order neural network (HONN) can be designed to rapidly learn to classify objects, invariant to their scale, angular orientation, and translational position. Other neural networks must be shown a very large number of distorted versions of a pattern in order to achieve distortion invariance. A properly designed HONN must only be trained on one example of each object to be learned. No further training is required to achieve invariant recognition, as distortion invariant feature extraction is built-in as part of the architecture.

We demonstrate a second-order network which performs pattern recognition with 100% accuracy for any translational position and over a scale factor of 5. The use of a third-order network to achieve simultaneous rotation, scale and position invariance is described. Because of their ability to efficiently perform both mappings required for pattern recognition applications, namely feature extraction and object classification, HONNs are superior to multi-level, first-order networks trained by back-propagation for distortion invariant pattern recognition.

Introduction

Pattern recognition may be viewed as a two part process of feature extraction followed by object classification[1-4]. First, a preliminary mapping from an image to a representation space is made, generally resulting in a significant degree of data reduction. A second mapping then operates on this reduced data to produce a classification or estimation in an interpretation space. Historically, these steps have required either mathematical mappings operating directly on a detected image [1,2] or initial feature extraction performed through optical processing followed by some form of analytical discrimination[3].

Both mappings may also be performed using neural network models[4]. In this paper we discuss neural networks both as classifiers in hybrid systems and as implementations of the complete pattern recognition operation. Emphasis is given to recognition invariant to distortions in scale, translational position and angular orientation. The relatively poor results with neural models performing the complete mapping from image to interpretation is attributable to the unsuitability of the models used for distortion invariant feature extraction. In contrast, higher-order neural networks can be designed to implement the extraction of simple but effective features suitable for in-plane distortion invariance. Simulation results of higher-order neural networks demonstrating simultaneous invariance to scale and translation will be presented.

Neural Networks for Pattern Recognition

Pattern recognition requires the nonlinear separation of pattern space into subsets representing the objects to be identified. Early research into neural networks concentrated on defining their potential for nonlinear discrimination[5,6]. It was found that a single layer, first-order neural network can only perform linear discrimination. However, either multilayer, first-order networks or single layer networks of higher order can provide the desired nonlinear separation[6].

The capability of neural networks to perform nonlinear separation can be applied both to extract image features and to interpret images based on a feature set. Practical applications in distortion invariant pattern recognition have been found for hybrid systems utilizing neural networks for classification. Troxel et. al.[7] successfully applied a multi-layer perceptron neural network trained with a backward error propagation (back-propagation) learning algorithm [8,9] to classify laser radar images of targets, invariant to position, rotation and scale. The data was first mapped into the magnitude of the Fourier transform with log radial and angle axis, $|F(\ln r, \theta)|$, feature space. Glover [10] describes a practical product-inspection system based on the optical Fourier transform and neural classification. Rotation and scale invariance has also been described in a system using complex-log conformal mapping combined with a distributed neural associative memory[11]. In all of these approaches utilizing neural classification, distortion invariance is achieved through non-neural feature extraction techniques.

It has been argued that nonlinear neural computing is theoretically superior to methods such as matched filters or linear correlation for the complete pattern recognition operation, including feature extraction[12]. However, the performance of neural networks to date fails to fulfill this promise. For instance, several types of neural associative memories have been shown to be computationally more expensive than matched filters in a study involving the recognition of line segments[13]. Multi-layer networks trained by back-propagation have also been applied to recognition tasks, examples being sonar signal classification [14] and distortion invariant character recognition[15,16]. In these cases, the networks achieved ≈ 80-90% recognition accuracy only after being shown a training set of images several hundred [14] or thousand [15,16] times. Learning by back-propagation to distinguish a "T" from a "C", invariant to translation and rotation, required over 5000 presentations of an exhaustive training set[15]. Learning to distinguish 36 patterns in a 5 x 5 pixel array invariant to translation required over a 1000 training set presentations to a network composed of two-layers, each with 25 Adelines arranged in slabs[16].

The relatively poor performance of neural networks in the preceding examples, most particularly the failure to produce efficient distortion invariant recognition, is due to the fact that first-order networks are poorly suited for extracting distortion invariant features. One layer of a typical first-order network is shown in Figure 1.

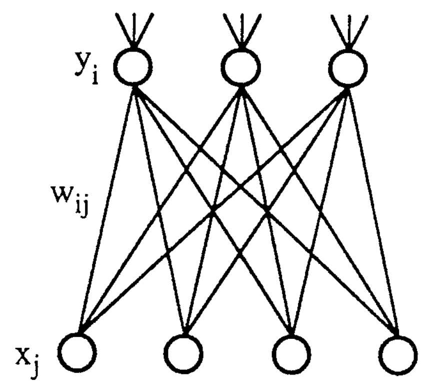


Figure 1: One layer of a first-order neural network.

The activation level of an output node in a first-order neural network is determined by an equation of the form:

$$y_i = \Theta(\Sigma_j\, w_{ij}\, x_j) \qquad (1)$$

where Θ is a nonlinear threshold function, the x_j are the excitation values of the input nodes, and the interconnection matrix elements, w_{ij}, determine the weight that each input is given in the summation.

Achieving translation, scale and rotation invariance requires a neural network to learn relationships between the input pixels, x_j. Note that the summation within the parenthesis in Eq. (1) is a function of individual x_j's. No advantage is taken of any known relationships between the x_j's. Multi-layer, first-order networks can learn invariances, but require a great deal of training, and produce solutions that are specific to particular training sets.

A further disadvantage is that the mappings learned are opaque: it is not readily evident what features are being extracted or how classification is being performed. It is generally assumed that the output of intermediate-layer hidden nodes in the network correspond to specific features, and in some applications it is possible to discern what these features are[14]. In distortion invariant recognition applications, however, it is not apparent that first-order networks' hidden nodes come to represent efficient feature sets or even feature sets sufficient to allow classification by succeeding layers.

Higher-Order Neural Networks

The output of nodes in a general higher order network is given by:

$$y_i = \Theta(\Sigma_j w_{ij} x_j + \Sigma_j \Sigma_k w_{ijk} x_j x_k + \Sigma_j \Sigma_k \Sigma_l w_{ijkl} x_j x_k x_l + \ldots) \quad (2)$$

A diagram of a neural network utilizing only second-order terms is shown in Figure 2. Higher-order neural networks (HONNs) were evaluated in the 1960s for performing nonlinear discrimination but were rejected as impractical due to the combinatoric explosion of higher-order terms[6].

Recent research [17-19] has shown that the problem of combinatoric explosion can be overcome by building invariances into the network architecture using information about the relationships expected between the input x_j's.

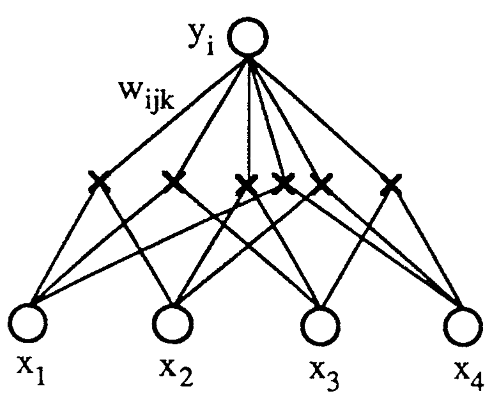


Figure 2: A second-order neural network with 4 inputs and 1 output.

HONNs are thus well suited for invariant pattern recognition because feature extraction is functionally built into the architecture. The invariances achieved require no learning to produce and apply to any input pattern learned by the network. Further, a HONN can perform nonlinear discrimination using only a single layer so that a simple perceptron learning rule can be used, leading to rapid convergence[4].

As an example, translation invariance can be built into the second-order neural network with 4 input nodes and 1 output node shown in Figure 2. Assume that the input patterns (1 0 1 0) and (0 1 0 1) are to be indentified as the same object. If $w_{i13} = w_{i24}$, then y_i is the same for both inputs. In general, translation invariance requires that:

$$w_{ijk} = w_{i(j-k)} \quad (3)$$

i.e., the connections for equally spaced input pairs are all set equal.

Combinations of invariances can similarly be achieved. A second-order neural network will be simultaneously invariant to scale and translation if the weights are set according to the function[18]

$$w(i,j,k) = w(i,(y_k - y_j)/(x_k - x_j)) \quad (4)$$

Equation (4) implies that w_{ijk} is set equal to $w_{ij'k'}$ if the slope of a line drawn between nodes j and k equals that formed between j' and k', as shown in Figure 3.

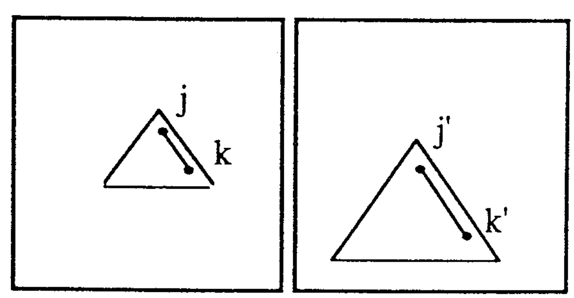


Figure 3: Translation and scale invariance achieved by setting $w_{ijk} = w_{ij'k'}$ if the slope of the line formed by nodes j and k equals that formed by nodes j' and k'.

Any object drawn in a 2-D plane can have lines of various slopes drawn within it. An object's relative content of lines of different slopes does not change when it is translated in position or scaled in size, as long as it is not rotated.

Rotational invariance can be included by using a third-order neural network, where the output is given by the function

$$y_i = \Theta(\Sigma_j \Sigma_k \Sigma_l \; w_{ijkl} \; x_j x_k x_l) \qquad (5)$$

As shown in Figure 4, any three points within an object define a triangle with included angles (α,β,γ). When the object is translated, scaled and rotated, the three points in the same relative positions on the object still form the included angles (α,β,γ). Therefore, invariances to all three distortions can be achieved with a third-order network having an interconnection function of the form:

$$w_{ijkl} = w_{i\alpha\beta\gamma} = w_{i\gamma\alpha\beta} = w_{i\beta\gamma\alpha} \quad (6)$$

Note that the order of angles matters, but not which angle is measured first.

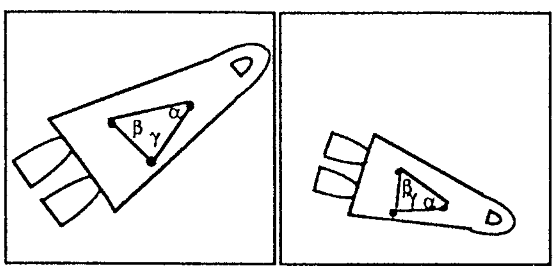


Figure 4: Translation, scale and rotation invariance is achieved by setting all third order weights equal for sets of inputs j, k, and l which form similar triangles.

Simulation results

We have simulated a second-order neural network to achieve simultaneous invariance to translation and scale. The single layer, second-order network is simulated using a 16 x 16, or 256 node, input field fully interconnected to a single output node which is thresholded with a fixed-threshold hard limiter:

$$\begin{aligned} \Theta(\Sigma) &= 1, \;\; \text{if } \Sigma > 0, \\ \Theta(\Sigma) &= -1, \;\; \text{if } \Sigma < -0 \end{aligned} \qquad (7)$$

There are 256-choose-2 or 32,640 input pairs and therefore interconnections. The interconnection weights are constrained to follow Eq. (4) in order to achieve invariance to scale and translation. The weights are initially set to zero and a learning rule is used of the form:

$$\Delta w_{ijk} = (t_i - y_i) \, x_j \, x_k \qquad (8)$$

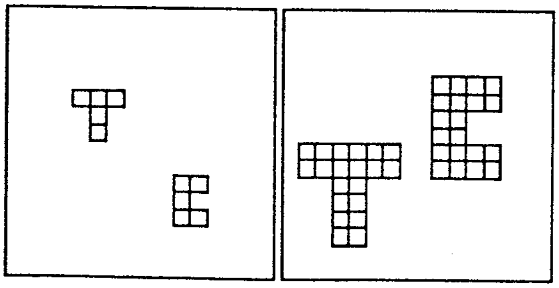
where the expected training output, t, actual output, y, and inputs x, are all binary. The network is trained on just 2 distinct patterns -- only one size and one location for each pattern. It learns to distinguish between the patterns after approximately 10 passes of the training set, requiring less than 1 minute of run time on a Sun 3 workstation. After training, it successfully distinguishes between all translated and scaled versions of the two objects with 100% accuracy. No further training is required to achieve this invariance, as it is built into the architecture. The system can learn to distinguish between any two distinct patterns, and has been tested on a variety of problems, including the T-C problem[5]. Scale invariance of a factor of 5 has been demonstrated for this problem, with 100% recognition accuracy.

Due to the limited resolution of the finite 16 x 16 input window, residual scale variance can occur. (T,C) pairs are distinguished by their relative content of horizontal and vertical information. For the smallest (T,C) pair, shown in Figure 5a, the T has 3 input pair combinations arranged horizontally and 3 vertically, while the C has 2 arranged horizontally and 4 vertically. In the next larger scale of (T,C), shown in Figure 5b, the ratio of horizontal to vertical pixel pairs is 34:34 for the T and 26:42 for the C. It is therefore easier to distinguish between the smaller (T,C) pair based on their relative horizontal/vertical content. If the system is trained on the smaller set of letters, learning is not pushed to the point where larger versions can be recognized. In contrast, if large patterns are used for training, all smaller versions are subsequently recognized.

Residual scale variance can be eliminated by using bipolar training values and a modified threshold function such as,

$$\begin{aligned} \Theta(\Sigma) &= 1, \;\; \text{if } \Sigma > K, \\ \Theta(\Sigma) &= -1, \;\; \text{if } \Sigma < -K, \\ \Theta(\Sigma) &= 0, \;\; \text{otherwise,} \end{aligned} \qquad (9)$$

where K is some positive constant. Learning with a sufficiently large value for K forces the network



(a) (b)

Figure 5: Two different scales of T and C drawn in a 16 x 16 pixel window.

to make a greater distinction between the initial patterns, allowing easier discrimination between test patterns which are subsequently evaluated with a hard limiter. Training the network on the smallest (T,C) pair using a value of K = 1000 allows correct identification of all larger test versions, without greatly increasing the training time.

Conclusion

Our simulations have demonstrated that a second-order neural network can be rapidly trained to distinguish between two patterns regardless of their size and translational position. 100% recognition accuracy is achieved for several different training pattern pairs using a 16 x 16 input field size. Training requires only 10-20 presentations of just one example of each object to be learned. In addition, a third-order network architecture to produce simultaneous invariance to rotation has been described. Comparisons in terms of recognition accuracy and learning speed show HONNs to be vastly superior to multi-layer first-order networks trained by back-propagation for this application.

This superiority results from the HONN architecture's ability to perform simple, transparent feature extraction. These simple features, slopes between input pixel pairs in the case of the second-order network, and included angles between input pixel triplets for the third-order network, are sufficient to allow the network to rapidly learn to classify patterns. The provision of a transparent feature extraction mechanism allows a HONN to efficiently perform the complete mapping from image to intermediate feature space to interpretation space required for distortion invariant pattern recognition.

References

[1] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[2] Chen, C.H., Statistical Pattern Recognition. Washington: Hayden, 1973.

[3] Q. Tian, Y. Fainman, Z.H. Gu and S.H. Lee, "Comparison of Pattern Recognition Algorithms for Hybrid Processing," J. Optical Soc. of America A. Vol. 5, 1988, pp. 1655-1669.

[4] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks. Reading, MA: Addison-Wesley, 1989.

[5] F. Rosenblatt, Principles of Neurodynamics. New York: Spartan, 1962.

[6] M.L. Minsky and S. Papert, Perceptrons. Cambridge, MA: MIT Press, 1969.

[7] S.E. Troxel, S.K. Rogers, and M. Kabrisky, "The Use of Neural Networks in PSRI Target Recognition," Proc. IEEE Int. Conf. on Neural Networks, San Diego, California, July 24-27. Vol. 1, 1988, pp. 593-600.

[8] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University, 1974 (unpublished).

[9] D.E. Rumelhart, G.E. Hinton, and R.J.Williams, "Learning Internal Representations by Error Propagation," Parallel Distributed Processing. Vol. 1, Ch. 8, Cambridge, MA: MIT Press, 1986.

[10] D.E. Glover, "An Optical Fourier/Electronic Neurocomputer Automated Inspection System," Proc. IEEE Int. Conf. on Neural Networks, San Diego, California, July 24-27. Vol. 1, 1988, pp. 569-576.

[11] H. Wechsler and G.L. Zimmerman, "Invariant Object Recognition Using a Distributed Associative Memory," Neural Information Processing Systems. American Institute of Physics Conference Proceedings, 1988, pp. 830-839.

[12] H. Szu, "Three Layers of Vector Outer Product Neural Networks for Optical Pattern Recognition," Optical and Hybrid Computing. SPIE Vol. 634, 1986, pp. 312-330.

[13] P.M. Grant and J.P. Sage, "A Comparison of Neural Network and Matched Filter Processing for Detecting Lines in Images," Neural Networks for Computing. American Institute of Physics Conference Proceedings, 1986, pp. 194-199.

[14] R.P. Gorman and T.J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," Neural Networks. Vol. 1, 1988, pp. 75-89.

[15] D.E. Rumelhart, *op. cit.*, pp. 348-352.

[16] B. Widrow and R. Winter, "Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition," IEEE Computer Magazine. Vol. 21, March, 1988, pp. 25-39.

[17] C.L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," Applied Optics. Vol. 26, 1987, pp. 2972-4978.

[18] C.L. Giles, R.D. Griffin, and T. Maxwell, "Encoding geometric invariances in higher-order neural networks," Neural Information Processing Systems. American Institute of Physics Conference Proceedings, 1988, pp. 301-309.

[19] M.B. Reid, L. Spirkovska, and E. Ochoa, "Rapid Training of Higher-Order Neural Networks for Invariant Pattern Recognition," Proc. Joint Int. Conf. on Neural Networks, Washington, D.C., June 18-22, 1989.